## What is claimed is:

2 3 4 5 6 7 8 9 10 11 12 13	selecting an initial sequence of instructions for inclusion in a trace cache line;  determining a set of rename resources needed for said trace cache line on a perpacket basis;  adding one or more provisional instructions to said trace cache line to create a provisional trace cache line;  repeating said determining step for said provisional trace cache line;  comparing said set of rename resources needed for said provisional trace cache
4 5 6 7 8 9 10 11 12	determining a set of rename resources needed for said trace cache line on a perpacket basis;  adding one or more provisional instructions to said trace cache line to create a provisional trace cache line;  repeating said determining step for said provisional trace cache line;  comparing said set of rename resources needed for said provisional trace cache
5 6 7 8 9 10 11 12	packet basis;  adding one or more provisional instructions to said trace cache line to create a provisional trace cache line;  repeating said determining step for said provisional trace cache line;  comparing said set of rename resources needed for said provisional trace cache
6 7 8 9 10 11 12	adding one or more provisional instructions to said trace cache line to create a provisional trace cache line; repeating said determining step for said provisional trace cache line; comparing said set of rename resources needed for said provisional trace cache
7 8 9 10 11	provisional trace cache line;  repeating said determining step for said provisional trace cache line;  comparing said set of rename resources needed for said provisional trace cache
8 9 10 11 12	repeating said determining step for said provisional trace cache line; comparing said set of rename resources needed for said provisional trace cache
9 10 11 12	comparing said set of rename resources needed for said provisional trace cache
10 11 12	
11 12	
12	line to a rename capacity; and
	accepting said one or more provisional instructions for inclusion in said trace line
13	and repeating said adding step, or rejecting said one or more provisional instructions,
	based on said comparing step.
1	2. A method in accordance with claim 1, wherein:
2	said set of rename resources needed and said rename capacity include a source
3	parameter.

1

3.

A method in accordance with claim 1, wherein:

2		said set of rename resources needed and said rename capacity include a destination		
3	parameter.			
1	4.	A method in accordance with claim 1, wherein:		
2		said set of rename resources needed and said rename capacity include a line size		
3	parameter.			
1	5.	A method in accordance with claim 1, wherein:		
2		determining a set of rename resources needed on a per-packet basis excludes		
3	destinations subsequently over-written within the packet from said set of rename			
4	resources needed.			
1	6.	A method in accordance with claim 1, wherein:		
2		determining a set of rename resources needed on a per-packet basis excludes		
3	redunc	lant sources within the packet from said set of rename resources needed.		
1	7.	A method in accordance with claim 1, wherein:		
2		determining a set of rename resources needed on a per-packet basis excludes		
3	source	es created within said trace cache line.		
1	8.	A method in accordance with claim 1, wherein:		

2		selecting said initial sequence of instructions uses a worst case assumption of said
3	set of	f rename resources needed.
1	9.	A method in accordance with claim 1, wherein:
2		selecting said initial sequence of instructions includes tabulating a maximum rename
3	resou	rce cumulative total based on a plurality of instruction types.
1	10.	A method in accordance with claim 1, wherein:
2		selecting a number of provisional instructions is performed based on a difference
3	betwe	een said set of rename resources needed and said rename capacity.
1	11.	An apparatus for processing instructions in a superscalar microprocessor,
2	comp	orising:
3		an instruction stream with a plurality of instructions;
4		a trace cache line for receiving said instructions from said instructions stream;
5		a packetized instruction resource calculator for determining a set of rename
6	resou	rces needed for said instructions in said trace cache line;
7		an instruction adder, responsive to said packetized instruction resource calculator,
8	for ac	dding one or more instructions to said trace cache line from said instruction stream
9	while	e said set of rename resources needed is less than a rename resource capacity.

1

12.

An apparatus in accordance with claim 11, wherein:

said set of rename resources needed includes a source parameter. 2 13. An apparatus in accordance with claim 11, wherein: 1 said set of rename resources needed included a destination parameter. 2 An apparatus in accordance with claim 11, wherein: 14. said set of rename resources needed includes a line size parameter. 2 An apparatus in accordance with claim 14, wherein: 15. said set of rename resources needed includes a source parameter. 2 16. An apparatus in accordance with claim 15, wherein: 1 said set of rename resources needed includes a destination parameter. 2 An apparatus in accordance with claim 11, wherein: 17. 1 said packetized instruction resource calculator excludes destinations subsequently 2 over-written within said trace cache line from said set of resources needed. 3 An apparatus in accordance with claim 17, wherein: 18. 1 said packetized instruction resource calculator excludes redundant sources with 2 said trace cache line from said set of resources needed. 3

l	19.	An apparatus in accordance with ciaim 18, wherein:	
2		said packetized instruction resource calculator excludes sources created within	
3	said tr	race cache line.	
l	20.	An apparatus in accordance with claim 11, further comprising:	
2		a trace cache line initializer for initially loading said trace cache lien with an initial	
3	numb	er of instructions.	
1	21.	An apparatus in accordance with claim 20, wherein:	
2		said initial number of instructions is calculated as a fraction of said rename	
3	resource capacity.		
1	22.	A method of creating cache lines of instructions in a computer system,	
2	compi	rising:	
3		determining the number of instructions in the cache lines using a packetization of	
4	instru	ctions technique and a dynamic cache line size;	
5		matching said dynamic cache line size to a rename unit capacity.	
1	23.	A method in accordance with claim 22, wherein:	
2		matching said dynamic cache line size includes eliminating redundant register	
3	refere	nces within the cache lines.	